

**МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ  
РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА”**

**Кафедра «Информационные технологии»**

**Т. Н. ЛИТВИНОВИЧ**

**ПРИМЕНЕНИЕ VBA  
ДЛЯ РЕШЕНИЯ  
ПРИКЛАДНЫХ ЗАДАЧ**

**Учебно-методическое пособие по выполнению контрольной работы №2**

**Гомель 2016**

МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ  
РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА”

Кафедра «Информационные технологии»

Т. Н. ЛИТВИНОВИЧ

# ПРИМЕНЕНИЕ VBA ДЛЯ РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ

*Одобрено методической комиссией заочного факультета  
в качестве учебно-методического пособия  
по выполнению контрольной работы №2  
по дисциплине «Информатика»  
для студентов 1-го курса специальности  
«Промышленное и гражданское строительство»*

Гомель 2016

УДК 004.438(075.8)  
ББК 32.81  
Л64

Рецензент – доцент кафедры «Информационные технологии»  
канд. техн. наук *Ю. А. Пшеничнов* (УО «БелГУТ»)

**Литвинович, Т.Н.**

Л64 Применение VBA для решения прикладных задач: учеб.- метод. пособие по выполнению контрольной работы №2 / Т. Н. Литвинович ; М-во трансп. и коммуникаций Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель : БелГУТ, 2016. – 39 с.  
ISBN 978-985-554-526-3

Кратко изложены теоретические сведения и рассмотрены примеры выполнения заданий, входящих в контрольную работу.

Предназначено для выполнения контрольной работы №2 по дисциплине «Информатика» студентами I курса заочного факультета специальности «Промышленное и гражданское строительство».

**УДК 004.438(075.8)**  
**ББК 32.81**

**ISBN 978-985-554-526-3**

© Литвинович Т. Н., 2016  
© Оформление. УО «БелГУТ», 2016

## СОДЕРЖАНИЕ

1	Логические функции MS Excel .....	4
2	VBA: основы программирования в программах пакета MS Office 2007	7
2.1	Объектно-ориентированная технология программирования .....	7
2.2	Синтаксис VBA.....	8
2.3	Программирование линейных алгоритмов.....	12
2.3.1	Среда разработки .....	12
2.3.2	Модули .....	13
2.3.3	Процедуры-подпрограммы и процедуры-функции .....	13
2.3.4	Ввод данных .....	15
2.3.5	Вывод данных .....	16
2.3.6	Ввод (вывод) данных в ячейки рабочего листа .....	17
2.3.7	Создание линейной функции пользователя рабочего листа .....	18
2.4	Программирование разветвляющихся алгоритмов .....	19
2.4.1	Условная инструкция .....	19
2.4.2	Инструкция выбора варианта .....	21
2.4.3	Создание разветвляющейся функции пользователя рабочего листа...	22
2.4.4	Макросы.....	23
3	Расчет статически определимой двутавровой балки.....	24
4	Пример выполнения заданий контрольной работы.....	26
5	Требования к форматированию итогового отчета.....	32
	Список литературы.....	33
	Приложение А Сортамент двутавров ГОСТ 8239 (сокращенная версия)	34
	Приложение Б Титульный лист (образец).....	35
	Приложение В Бланк задания на контрольную работу № 2.....	36

## 1 ЛОГИЧЕСКИЕ ФУНКЦИИ MS EXCEL

Все функции в *MS Excel* делятся на категории. Стандартные математические и статистические функции были рассмотрены ранее. Предметом рассмотрения в этом разделе будут логические функции и их применение для решения задач.

Функция **ЕСЛИ** служит для выполнения вычисления по одной или другой формуле в зависимости от результата проверки некоторого условия и имеет следующий формат:

**ЕСЛИ (условие; [значение\_если\_истина]; [значение\_если\_ложь])**.

Здесь,

условие – логическое выражение, результатом вычисления которого может быть одна из двух констант **ИСТИНА** или **ЛОЖЬ**.

значение\_если\_истина – константа или формула, возвращающая некоторое значение, если результатом вычисления условия является значение **ИСТИНА**.

значение\_если\_ложь – константа или формула, возвращающая некоторое значение, если результатом вычисления условия является значение **ЛОЖЬ**.

Условия можно классифицировать на простые и составные. Для записи условий используют:

1) операции отношения: >, <, >=, <=, =, <> “не равно”.

2) логические функции: **И**, **ИЛИ**, **НЕ**. Служат для записи составных условий.

Функция **И** возвращает значение **ИСТИНА**, если результатом проверки всех входящих в нее условий является значение **ИСТИНА**, и значение **ЛОЖЬ** в противном случае. Данная функция имеет следующий формат:

**И (условие1; условие 2; [условие n])**.

Функция **ИЛИ** возвращает значение **ИСТИНА**, если результатом проверки хотя бы одного из заданных условий является значение **ИСТИНА**, и значение **ЛОЖЬ** в противном случае. Данная функция имеет следующий формат:

**ИЛИ (условие1; условие 2; [условие n]).**

Функция **НЕ** возвращает значение ИСТИНА, если результатом проверки условия является значение ЛОЖЬ и наоборот. Данная функция имеет следующий формат:

**НЕ (условие).**

Функция **ЕСЛИ** допускает запись и более сложных логических формул, в которых может осуществляться выбор более чем из двух альтернативных вариантов решения задачи. В этом случае она будет иметь такой формат и записывается в одну строку:

ЕСЛИ (условие1; значение1\_истина;  
ЕСЛИ (условие2; значение2\_истина;  
ЕСЛИ (условие n; значение n\_истина; значение\_ложь))).

*Примечание* – В квадратных скобках указаны не обязательные параметры.

Рассмотрим ряд примеров на применение логических функций.

**Пример 1.** Вычислить значение функции в заданной расчетной точке.

$$f(x) = \begin{cases} \sqrt{x-2} + x^2, & \text{если } x > 2 \\ \cos x, & \text{в остальных случаях} \end{cases}$$

Выбор формулы, по которой будет вычисляться функция, зависит от значения переменной  $x$ , которое нужно проверять на соответствие заданному условию. Следовательно, для решения задачи будем использовать логическую функцию ЕСЛИ. Результат решения задачи в режиме отображения формул показан на рисунке 1, а в режиме отображения значений на рисунке 2.

	A	B	C
1			
2		x	f(x)
3		3	=ЕСЛИ(B3>2;КОРЕНЬ(B3-2)+СТЕПЕНЬ(B3;2);COS(B3))
4		-4	=ЕСЛИ(B4>2;КОРЕНЬ(B4-2)+СТЕПЕНЬ(B4;2);COS(B4))
5			

Рисунок 1 – Результат примера 1 в режиме формул

	A	B	C
1			
2		x	f(x)
3		3	10
4		-4	-0,653643621
5			

Рисунок 2 – Результат примера 2 в режиме значений

**Пример 2.** Вычислить значение функции в заданной расчетной точке.

$$f(x) = \begin{cases} \sqrt{x-2} + x^2, & \text{если } x \geq 2 \\ \cos x, & \text{если } -10 \leq x < 2 \\ x^3, & \text{в остальных случаях} \end{cases}$$

Результат решения задачи в режиме отображения формул показан на рисунке 3, а в режиме отображения значений – на рисунке 4.

	A	B	C
1			
2		x	f(x)
3	3		=ЕСЛИ(В3>=2;КОРЕНЬ(В3-2)+СТЕПЕНЬ(В3;2);ЕСЛИ(И(В3>=-10;В3<2);COS(В3);СТЕПЕНЬ(В3;3)))
4	-4		=ЕСЛИ(В4>=2;КОРЕНЬ(В4-2)+СТЕПЕНЬ(В4;2);ЕСЛИ(И(В4>=-10;В4<2);COS(В4);СТЕПЕНЬ(В4;3)))
5	-15		=ЕСЛИ(В5>=2;КОРЕНЬ(В5-2)+СТЕПЕНЬ(В5;2);ЕСЛИ(И(В5>=-10;В5<2);COS(В5);СТЕПЕНЬ(В5;3)))
6			

Рисунок 3 – Результат примера 2 в режиме формул

	A	B	C
1			
2		x	f(x)
3		3	10
4		-4	-0,653643621
5		-15	-3375
6			

Рисунок 4 – Результат примера 2 в режиме значений

*Примечание* – Составное неравенство  $-10 \leq x < 2$  задается с помощью функции И (В3 >= -10; В3 < 2).

### Пример 3.

Покупатель магазина получает скидку 3%, если у него есть дисконтная карта или если общая стоимость его покупки превышает 300000 рублей. Сколько заплатили покупатели за свои покупки?

Результат решения задачи в режиме отображения формул показан на рисунке 5, а в режиме отображения значений на рисунке 6.

	A	B	C	D
1				
2	Ф. И.О. покупателя	Стоимость покупки, руб.	Дисконтная карта	Стоимость итого, руб.
3	Петров С.Н.	120000	да	=ЕСЛИ(ИЛИ(В3>300000;С3="да");В3-В3*0,03;В3)
4	Сидоров Н.А.	150000	нет	=ЕСЛИ(ИЛИ(В4>300000;С4="да");В4-В4*0,03;В4)
5	Алексеева Н.И.	310000	нет	=ЕСЛИ(ИЛИ(В5>300000;С5="да");В5-В5*0,03;В5)
6	Грушин А.К.	340000	да	=ЕСЛИ(ИЛИ(В6>300000;С6="да");В6-В6*0,03;В6)
7	Сергина Р.Н.	160000	нет	=ЕСЛИ(ИЛИ(В7>300000;С7="да");В7-В7*0,03;В7)
8	Крупцин Е.Н.	315000	нет	=ЕСЛИ(ИЛИ(В8>300000;С8="да");В8-В8*0,03;В8)
9				

Рисунок 5 – Результат примера 3 в режиме формул.

	A	B	C	D	E
1					
2	Ф. И.О. покупателя	Стоимость покупки, руб.	Дисконтная карта	Стоимость_итого, руб.	
3	Петров С.Н.	120 000,00р.	да	116400	
4	Сидоров Н.А.	150 000,00р.	нет	150000	
5	Алексеева Н.И.	310 000,00р.	нет	300700	
6	Грушин А.К.	340 000,00р.	да	329800	
7	Серегина Р.Н.	160 000,00р.	нет	160000	
8	Крупицин Е.Н.	315 000,00р.	нет	305550	

Рисунок 6 – Результат примера 3 в режиме значений

## 2 VBA: ОСНОВЫ ПРОГРАММИРОВАНИЯ В ПРОГРАММАХ ПАКЕТА MS OFFICE 2007

### 2.1 Объектно-ориентированная технология программирования

В состав пакета *MS Office 2007* входит большое количество приложений: MS Excel, MS Word, MS Access и др. Для расширения функциональности этих приложений, а также для обеспечения совместной работы с ними разработчиками вышеуказанного пакета был предложен язык программирования Visual Basic for Applications (VBA) встроенный в Microsoft Office. Данный язык программирования является объектно-ориентированным. Ключевым понятием объектно-ориентированной технологии программирования является понятие “**объект**”. Объект – это программная единица, состоящая из данных и процедур работы с ними. Данные описывают характеристики объекта, а процедуры – действия, которые можно с ним производить. Данные называют свойствами объекта, а процедуры его методами.

**Свойства** объекта описывают его характеристики.

Например, для того, чтобы применить к выделенному фрагменту текста в *Word* цвет синий, начертание – курсив, надо установить соответствующие свойства, используя в программе следующие команды:

*Selection.Font.ColorIndex = wdBlue*

*Selection.Font.Italic = True*

Здесь

*Selection* – объект (выделенный фрагмент);

*Font* – объект (шрифт);

*ColorIndex* – свойство объекта «Шрифт», устанавливающее цвет;

*Italic* – свойство объекта «Шрифт», устанавливающее курсивное начертание;

*wdBlue* – значение свойства «Цвет» (встроенная константа, устанавливающая синий цвет текста);

*True* – значение свойства «Курсивное начертание» (логическая констан-



та, означающая, что курсивное начертание установлено).

Таким образом, синтаксис установки значения свойства можно описать следующей формулой:

`Название_объекта.Свойство = Значение_свойства`

Например, чтобы изменить ширину 2 столбца в *Excel*, надо установить соответствующее свойство, используя в программе следующую команду:

*Columns("B:B").ColumnWidth = 30*

Здесь,

*Columns("B:B")* – объект (первый столбец таблицы);

*ColumnWidth* – свойство (ширина столбца), значение которого устанавливается равным 30.

*Примечание* – Установку свойств объекта можно производить динамически, прописывая соответствующую команду в программе, или статически в окне свойств *Properties* редактора *VBE*.

**Методы** объекта описывают действия, которые можно над ним выполнять.

Например, чтобы удалить из ячейки g2 второго листа рабочей книги содержимое необходимо написать следующую команду:

*Worksheets(2).Range("g2").ClearContents*

Здесь,

*Worksheets(2)* – объект (рабочий лист 2 активной рабочей книги);

*Range("g2")* – объект (ячейка g2);

*ClearContents* – метод (очистить содержимое ячейки).

**Например**, чтобы удалить список в *Word* необходимо написать следующую команду:

*Selection.Range.ListFormat.RemoveNumbers*

Здесь,

*Selection* – объект (выделение);

*ListFormat* – объект (список);

*RemoveNumbers* – метод (удалить нумерацию в списке).

Таким образом, синтаксис применения метода можно описать следующей формулой:

`Название_объекта.Метод`

## 2.2 Синтаксис VBA

**Алфавит** – это полный набор допустимых символов, принятых в конкретном языке. Алфавит языка *VBA* образуют:

– прописные и строчные буквы латинского алфавита;

- десятичные цифры;
- знак подчеркивания «\_»;
- специальные символы: +; -; \*; /; =; >; <; [ ]; ( ); { } и т.п.

Знаки алфавита используются для составления слов. Слова – неделимые последовательности знаков алфавита, несущие определенный смысл. Слова подразделяются на две группы: ключевые и идентификаторы пользователя.

Ключевые слова применяются для обозначения системных переменных, констант, процедур, функций и различных объектов;

Идентификаторы пользователя применяются для обозначения переменных, констант, процедур, функций и различных объектов, определенных самим пользователем.

Любой объект программы (переменная или константа) относится к какому-либо **типу данных**. Тип данных определяет:

- диапазон возможных значений;
- перечень допустимых операций;
- объем выделяемой памяти и форму представления данных в ней.

Базовые типы данных *VBA* приведены в таблице 1.

Таблица 1 – Типы данных

Тип данных	Описание и диапазон значений
<i>Byte</i>	Положительное число от 0 до 255
<i>Integer</i>	Целое число от -32 768 до 32 767
<i>Long</i>	Длинное целое число от -2 147 483 648 до 2 147 483 647
<i>Single</i>	Число с плавающей точкой обычной точности. Отрицательные числа от -3,402823E+38 до 1,401298E-45. Положительные числа от 1,401298E-45 до 3,402823E+38
<i>Double</i>	Число с плавающей точкой двойной точности. Отрицательные числа: от -1,797693E+308 до -4,940656E-324. Положительные числа: 4,940656E-324 до 1,797693E+308
<i>Boolean</i>	<i>True</i> (истина) или <i>False</i> (ложь)
<i>Variant</i>	Может использоваться для хранения практически любого типа данных. Диапазон зависит от фактически сохраняемых данных

**Переменными** называются значения в процедуре, которые в процессе ее выполнения могут изменяться. Для объявления переменных в процедуре используется следующая схема:

*Dim* имя\_переменной [*as* тип]

**Например,**

*Dim a as integer*

*Dim b,c as single*

**Константами** называются значения в процедуре, которые в процессе ее выполнения не изменяются.

Константы делятся на два типа:

1 Литеральные. Используются в выражениях.

$$a = b + 3.$$

2 Именованные, которые делятся на системные и пользовательские. Пользовательские константы предварительно объявляются перед их использованием в процедуре по следующей схеме:

`Const имя_константы [as тип] = значение`

Например,

`Const pi as single = 3.14`

Системные константы заранее определены.

Константы в зависимости от типа данных бывают:

1 числовые: целые и вещественные;

2 строковые. Записываются в кавычках “”;

3 даты. Записываются между символами ##;

4 логические. Принимают одно из двух значений *True* (Истина) или *False* (Ложь).

Все **операции VBA** можно разделить на следующие группы:

1 Арифметические. Выполняются над числами. Результат выполнения таких операций – числовое значение.

2 Отношения. Служат для выполнения сравнений. Результат выполнения таких операций – одно из двух логических значений *True* или *False*.

3 Логические. Служат для выполнения сравнений. Результат выполнения таких операций – одно из двух логических значений *True* или *False*.

4 Конкатенации (&). Служит для объединения нескольких строк в одну.

К арифметическим операциям относят: сложение (+), вычитание (–), умножение (\*), деление (/), возведение в степень (^), остаток от деления (mod), целая часть от деления (\).

К операциям отношения относят: >, <, >=, <=, =, <> «не равно».

К логическим операциям относят: *and* (логическое умножение), *or* (логическое сложение), *not* (логическое отрицание).

**Выражение** – это комбинация операндов, круглых скобок и знаков операций.

**Операнд** – объект выражения, над которым выполняется действие.

Различают числовые, строковые, логические выражения.

Числовое выражение – это любое выражение, значением которого является число.

$$c = \frac{a + b}{\sin(a^2)}$$

Строковое выражение – любое выражение, значением которого является последовательность символов.

`c = “Сегодня” & “8 марта”`

Логическое выражение – любое выражение, возвращающее логическое значение Истина (*True*) или Ложь (*False*).

$$c = (a > 2) \text{ and } (b > 2) \text{ and } (d > 2)$$

Порядок выполнения операций (см. таблицу 2) зависит от их приоритета.

Таблица 2 – Приоритет выполнения операций

Приоритет	Операция
1	Вызов функции и скобки
2	^
3	- (смена знака)
4	*, /
5	\
6	<i>Mod</i>
7	+, -
8	>, <, >=, <=, <>, =
9	<i>Not</i>
10	<i>And</i>
11	<i>Or</i>

В VBA существует большое количество **встроенных функций**, которые делятся на категории. Основные математические функции представлены в таблице 3.

Таблица 3 – Математические функции VBA

Функция	Возвращаемое значение
<i>Abs</i> (число)	Модуль (абсолютная величина) числа
<i>Atn</i> (число)	Арктангенс числа
<i>Cos</i> (число)	Косинус числа
<i>Sin</i> (число)	Синус числа
<i>Exp</i> (число)	Экспонента, т. е. результат возведения основания натурального логарифма в указанную степень
<i>Log</i> (число)	Натуральный логарифм числа
<i>Rnd</i> (число)	Случайное число из интервала [0,1). Перед вызовом функции <i>Rnd</i> необходимо использовать функцию <i>Randomize()</i> без аргумента
<i>Sgn</i> (число)	Знак числа
<i>Sqr</i> (число)	Квадратный корень из числа
<i>Tan</i> (число)	Тангенс числа
<i>Fix</i> (число) <i>Int</i> (число)	Обе функции отбрасывают дробную часть числа и возвращают целое значение.

## 2.3 Программирование линейных алгоритмов

Решение задач с помощью VBA требует создания проекта.

Проект – это совокупность нескольких элементов:

- 1) приложения (MS Excel, MS Word и др.);
- 2) среды разработки;
- 3) модулей.

### 2.3.1 Среда разработки

Среда разработки используется для разработки проектов на языке VBA, и запускается при помощи комбинации клавиш *Alt+F11* или командой Visual Basic, находящейся на вкладке Разработчик. Она состоит из редактора Visual Basic (Visual Basic Editor, VBE), средств отладки, средства управления проектами и т. д. Внешний вид среды разработки, с указанием некоторых базовых элементов показан на рисунке 7.

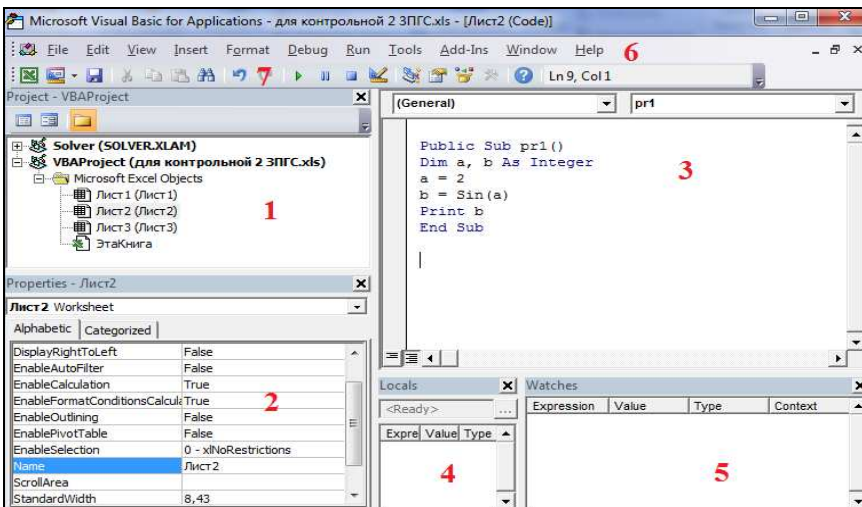


Рисунок 7 – Среда разработки

На рисунке 7 показаны следующие элементы среды разработки:

1 Окно проекта (*Project Explorer*). Здесь указан перечень элементов, входящих в текущий проект. По умолчанию создаются модули для трех рабочих листов и самой рабочей книги. Это окно вызывается на экран при помощи комбинации клавиш *Ctrl+R*.

2 Окно свойств (*Properties Window*). Здесь настраиваются свойства объектов, входящих в проект. Можно настроить, например, свойства каждого рабочего листа и самой рабочей книги в целом. Это окно вызывается на экран при помощи функциональной клавиши *F4*.

3 Окно редактора программного кода (*Code*). Здесь пользователь формирует процедуры и функции, относящиеся к соответствующему модулю. Это окно вызывается на экран при помощи функциональной клавиши *F7*.

4 Окно просмотра текущих значений переменных и выражений в процессе пошагового выполнения процедуры (*Locals Window*).

5 Окно контролируемых значений переменных и выражений (*Watches*). Позволяет самому пользователю задать переменные и выражения, значения которых он будет контролировать в процессе выполнения той или иной процедуры.

6 Главное меню, содержащее перечень всех команд, используемых при работе со средой программирования.

7 Панель инструментов, содержащая кнопки для быстрого выполнения той или иной команды.

### 2.3.2 Модули

Модули делятся на три вида.

1 Модули класса. Содержат описания свойств, методов и событий создаваемых пользователем классов.

2 Стандартные модули. Содержат доступные для создаваемого проекта процедуры.

Стандартные модули состоят из одной или нескольких процедур. Все процедуры делятся на два вида:

– процедуры-подпрограммы *Sub* – служат для выполнения заданной последовательности инструкций, приводящих решению задачи.

– процедуры-функции *Function*. Отличительной особенностью этого вида процедур является то, что результатом их работы всегда является некоторое значение.

3 Модули форм. Содержат процедуры обработки событий, генерируемые элементами управления форм создаваемого проекта.

### 2.3.3 Процедуры-подпрограммы и процедуры-функции

Описание процедуры-подпрограммы осуществляется по следующей схеме:

```
Sub имя_процедуры([формальные_параметры])  
    тело_процедуры
```

```
End Sub
```

Здесь

имя\_процедуры – правильный идентификатор VBA;

формальные\_параметры – список параметров, которые передаются процедуре при ее вызове. Параметры отделяются друг от друга с помощью запятой. Возможно объявление процедуры без параметров;

тело\_процедуры – инструкции VBA, приводящие к решению поставленной задачи.

Вызов одной процедуры из другой процедуры проекта осуществляется при помощи следующей команды:

*Call* имя\_процедуры [список фактических параметров].

При этом происходит передача значений фактических параметров соответствующим формальным параметрам. Под фактическим параметром понимается константа, переменная или выражение, которое передается в процедуру при ее вызове.

Передача параметров может осуществляться одним из двух способов:

– по ссылке. Ключевое слово *ByRef* записывается перед именем формального параметра означает, что фактический параметр передается по ссылке, и его значение может изменить вызываемая процедура в процессе своего выполнения.

– по значению. Ключевое слово *ByVal* записывается перед именем формального параметра означает, что фактический параметр передается по значению, которое процедура не может изменить в процессе своего выполнения.

*Примечание* – В VBA фактические параметры по умолчанию передаются по ссылке.

Описание процедуры-функции осуществляется по следующей схеме:

*Function* имя\_функции([формальные\_параметры]) as [тип\_результата]  
тело\_функции

*End Function*

Здесь

имя\_функции – правильный идентификатор VBA;

тело\_функции – инструкции VBA, приводящие к решению поставленной задачи. Как правило, последней в теле функции должна быть такая инструкция: имя\_функции = выражение;

выражение – арифметическое, логическое или строковое выражение, которое вычисляется данной функцией.

**Программный код процедуры формируется согласно следующим правилам:**

1 В процедуре необходимо выделять описательную и исполнительную части.

2 Описательная часть содержит описания констант, переменных, массивов и других объектов, которые будут использованы в исполнительной части процедуры. Смысл описания указанных выше объектов состоит в выделении памяти для дальнейшей работы с ними.

3 Исполнительная часть содержит инструкции VBA, приводящие к решению поставленной задачи.

4 Для пояснения к фрагменту программного кода процедуры используются комментарии. Для их ввода используется символ (^) апостроф.

5 Длинные инструкции можно размещать в нескольких строках. Для этого используют признак продолжения строки, состоящей из двух символов: пробела и подчеркивания(\_).

6 Строка программного кода не может состоять более чем из 1024 символов.

7 Для разделения нескольких инструкций записанных в одну строку применяется двоеточие «:».

Вызов функции осуществляется так:

имя\_функции ([фактические параметры])

**Линейный алгоритм** – это такой алгоритм, в котором все действия выполняются последовательно друг за другом и только один раз.

Для программирования таких алгоритмов используют инструкцию присваивания, а также диалоговые окна для ввода (вывода) данных.

**Инструкция присваивания** осуществляет присваивание результата вычисления некоторого выражения переменной или константе и имеет следующий вид:

<идентификатор> = <выражение>

Для организации ввода (вывода) данных используются два вида диалоговых окон: *окна сообщений* и *окна ввода*.

### 2.3.4 Ввод данных

Функция ***InputBox*** служит для ввода исходных данных необходимых для решения задачи. Она имеет следующий формат:

*InputBox* (сообщение [, заголовок окна] [, умолчание])

Здесь

сообщение – строка, поясняющая данные, которые будет вводить пользователь в соответствующее текстовое поле;

заголовок окна – текст, размещаемый в строке заголовка диалогового окна. Если этот аргумент опущен, то в строку заголовка помещается имя приложения;

умолчание – константа, используемая в качестве исходного данного, если ничего не было введено в текстовое поле.

Функция вызывает на экран диалоговое окно (см. рисунок 8), содержащее текстовое поле для ввода информации и две кнопки *OK* и *Cancel*.

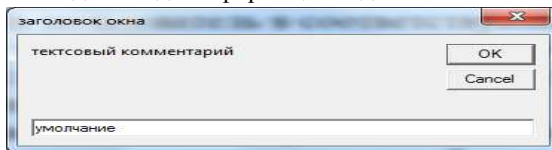


Рисунок 8 – Окно ввода данных



После ввода информации пользователем в текстовое поле и нажатии на кнопку *OK* возвращается результат, являющийся содержимым текстового поля. Если пользователь ничего не ввел и нажал кнопку *OK*, то возвращается значение, которое было задано в текстовом поле по умолчанию. При нажатии на кнопку *Cancel* функция ничего не возвращает.

Для того чтобы информация, введенная в текстовое поле, была сохранена в некую переменную, в программе используют оператор присваивания.

Например:

*b=inputbox*("введи значение b", "ввод данных")

*Примечание* – Информация, вводимая в текстовое поле, по умолчанию воспринимается как текст. При решении инженерных задач вводить в основном требуется числовые данные. Для того, чтобы преобразовать текст в число используется функция преобразования формата *Val()*. Кроме того, в VBA имеются также функции преобразования в числовое значение конкретного типа: *Cint()* – в целое число, *CSng()* – в вещественное число одинарной точности.

### 2.3.5 Вывод данных

Процедура *MsgBox* служит для организации вывода информации, в частности для организации вывода результатов решения задачи.

Она имеет следующий формат:

*MsgBox* сообщение [, атрибуты] [, заголовок окна]

Здесь,

сообщение – строка, отображаемая в диалоговом окне.

заголовок – строка, отображаемая в заголовке диалогового окна.

атрибуты – числовое выражение из констант VBA, которые задают количество и тип отображаемых в окне кнопок (см. таблицу 4), а также тип значка к сообщению (таблица 5). Например, чтобы вывести информационный значок и кнопку *OK*, необходимо ввести такое выражение *0+64* или *vbOKOnly+ vbInformation*.

Таблица 4 – Константы, описывающие количество и тип кнопок

Константа	Значение	Описание
<i>vbOKOnly</i>	0	Отображается только кнопка <i>OK</i>
<i>vbOKCancel</i>	1	Отображаются кнопки <i>OK</i> и " <i>Отмена</i> "

Окончание таблицы 4

<i>vbAbortRetryIgnore</i>	2	Отображаются кнопки " <i>Прервать</i> ", " <i>Повторить</i> " и " <i>Пропустить</i> "
<i>vbYesNoCancel</i>	3	Отображаются кнопки " <i>Да</i> ", " <i>Нет</i> " и " <i>Отмена</i> "
<i>vbYesNo</i>	4	Отображаются кнопки " <i>Да</i> " и " <i>Нет</i> "
<i>vbRetryCancel</i>	5	Отображаются кнопки " <i>Повторить</i> " и " <i>Отмена</i> "

Таблица 5 – Константы, описывающие тип значка

Константа	Значение	Описание
<i>vbCritical</i>	16	Отображается значок сообщения о критической ошибке.
<i>vbQuestion</i>	32	Отображается значок предупреждения с запросом.
<i>vbExclamation</i>	48	Отображается значок предупреждения.
<i>vbInformation</i>	64	Отображается значок информационного сообщения.

Данная процедура выводит на экран диалоговое окно, содержащее *сообщение*.

*Сообщение* может состоять из нескольких частей, которые связывают между собой знаком **&**. Сообщение может содержать строковые константы, записываемые в двойных кавычках "", имена переменных, констант, элементов массивов, а также вызовы функций. Для формирования сообщения, состоящего из нескольких строк, используют функцию перехода на новую строку *Chr(13)*.

**Задача.** Вычислить объем цилиндра, если известен радиус его основания *R* и высота *H*.

```
Public Sub pr1()
Const pi As Single = 3.14
Dim r, h, v As Single
'ввод данных
r = CSng(InputBox("введи радиус", "ввод данных", 3))
h = CSng(InputBox("введи высоту", "ввод данных", 5))
'расчет
v = pi * r ^ 2 * h
'результат
MsgBox "исходные данные: " & Chr(13) & "r=" & r & "h=" & h &
Chr(13) & "объем: " & v, vbInformation + vbOKOnly, "вывод результа-
тов"
End Sub
```

*Примечание* – Процедура *msgbox* в программном коде выше записывается в одну строку.

### 2.3.6 Ввод (вывод) данных в ячейки рабочего листа

Ввод и вывод данных можно также осуществлять, используя ячейки рабочей книги. Для этого используются следующие команды:

- **Range** (“Диапазон ячеек”);
- **Cells** (номер строки, номер столбца).*Value*.

**Задача.** Вычислить объем цилиндра, если известен радиус его основания *R* и высота *H*. Исходные данные считывать из ячеек *a2* и *b2* первого рабочего листа, а результаты расчета выводить в ячейку *c2* того же листа.

```
Public Sub pr2()
Const pi As Single = 3.14
Dim r, h, v As Single
'ввод данных
r = Worksheets(1).Range("a2").value
h = Worksheets(1).Cells("b2").value
'расчет
```

```
v = pi * r ^ 2 * h
'результат
Worksheets(1).cells(2,3).value = v
End Sub
```

*Примечание* – Программный код этой процедуры записывается в модуле текущей рабочей книги. Если размещать эту процедуру в модуле листа 1, тогда ссылку на объект “рабочий лист 1” *Worksheets(1)* при вводе и выводе данных делать не нужно.

### 2.3.7 Создание линейной функции пользователя рабочего листа

При работе в MS Excel для набора сложных формул применяется «Мастер функций», который в распоряжение пользователя предоставляет огромное количество заранее определенных встроенных в программу функций. Пользователю остается только выбрать необходимую функцию из перечня и задать для нее аргументы, прочитав их значения из соответствующих ячеек рабочего листа. VBA предоставляет пользователю инструменты для создания собственной функции, вызываемой через «Мастер функций». Такая функция должна находиться в отдельном стандартном модуле, создаваемом в среде VBA командой *Insert-Module*. В окне «Мастера функций» она будет находиться в категории «Определенные пользователем».

**Задача.** Создать функцию пользователя, позволяющую вычислять объем цилиндра, если известен радиусом его основания  $R$  и высота  $H$ . Затем вычислить объемы нескольких цилиндров радиусами  $2 \leq R \leq 12$   $\Delta R = 0.8$  и высотой  $H=5$  на листе 1 текущей рабочей книги.


#### Порядок решения задачи:

1 Запустить редактор VBE, нажав *Alt+F11* или кнопку *Visual Basic* на вкладке Разработчик.

2 Создать новый модуль, выполнив команду *Insert – Module*.

3 Сформировать программный код функции

```
Public Function V(R As Single, H As Single) As Single
V = 3.14 * R ^ 2 * H
End Function
```

4 Вернуться в окно программы *MS Excel*, нажав кнопку  на панели инструментов.

5 Ввести исходные данные для решения задачи (значения  $R$  и  $H$ ).

	A	B
1	R	H
2	2	
3	2,8	
4	3,6	
5	4,4	
6	5,2	
7	6	
8	6,8	
9	7,6	
10	8,4	
11	9,2	5
12	10	
13	10,8	
14	11,6	

6 Вычислить значение объема для  $R=2$ . Для этого установить курсор в ячейку C2, вызвать «Мастер функций», выбрать категорию «Определенные пользователем», выбрать функцию с именем V, в появившемся окне указать адреса ячеек с исходными данными.

7 Вычислить остальные значения, используя маркер автозаполнения и абсолютную ссылку на ячейку со значением H. Результаты в режиме отображения формул и значений показаны ниже.

	A	B	C
1	R	H	V
2	2	5	62,79999924
3	2,8		123,0879974
4	3,6		203,4719849
5	4,4		303,9520264
6	5,2		424,5279541
7	6		565,2000122
8	6,8		725,9680176
9	7,6		906,8319702
10	8,4		1107,79187
11	9,2		1328,8479
12	10		1570
13	10,8		1831,248047
14	11,6		2112,592041

Режим значений

	A	B	C
1	R	H	V
2	2	5	=V(A2;\$B\$2)
3	=A2+0,8		=V(A3;\$B\$2)
4	=A3+0,8		=V(A4;\$B\$2)
5	=A4+0,8		=V(A5;\$B\$2)
6	=A5+0,8		=V(A6;\$B\$2)
7	=A6+0,8		=V(A7;\$B\$2)
8	=A7+0,8		=V(A8;\$B\$2)
9	=A8+0,8		=V(A9;\$B\$2)
10	=A9+0,8		=V(A10;\$B\$2)
11	=A10+0,8		=V(A11;\$B\$2)
12	=A11+0,8		=V(A12;\$B\$2)
13	=A12+0,8		=V(A13;\$B\$2)
14	=A13+0,8		=V(A14;\$B\$2)

Режим формул

## 2.4 Программирование разветвляющихся алгоритмов

Разветвляющийся алгоритм отличается от линейного алгоритма тем, что в нем, перед тем как выполнить некую последовательность действий, требуется предварительно проверить условие, и по результатам этой проверки принять соответствующее решение. Для программирования таких алгоритмов применяются две инструкции: условную и выбора варианта.

### 2.4.1 Условная инструкция

Существует в нескольких форматах.

Краткая форма однострочный формат:

<i>If</i> <условие> <i>Then</i> <инструкции>
--

Здесь

Условие – логическое выражение.

*If..Then* – ключевые слова, означающие дословно «Если..То».

Данная инструкция обеспечивает выполнение заданных инструкций, если условие выполняется, и переход к выполнению следующей инструкции в процедуре в противном случае.

Условия делятся на две категории:

1 Простые. Для их записи используют операции отношения:  $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $<>$ . Например,  $a > b$  или  $c < 2$ .

2 Составные. Состоят из нескольких простых, связанных логическими операциями *and*, *or*, *not*. Например, “значение переменной  $x$  принадлежит интервалу  $[-5;2]$ ”. В этом случае условие будет выглядеть так:  $(x >= -5)$  *and*  $(x <= 2)$

*Примечание* – Инструкции в данном формате записываются в одну строку и разделяются символом «двоеточие».

**Задача.** Вычислить  $y(x) = \ln(x-3) + \sin(x)$ . Задача будет иметь решения в том случае, если  $x > 3$ .

```
Public Sub primer()  
Dim x As Integer  
Dim y as Single  
x = Val(InputBox("введи x", "ввод данных"))  
If x > 3 Then y = Log(x - 3) + Sin(x): MsgBox "x=" & x & " y=" & y  
End Sub
```

Краткая форма блочный формат:

*If* <условие> *Then*

<инструкции>

*End if*

**Задача.** Вычислить  $y(x) = \ln(x-3) + \sin(x)$

```
Public Sub primer()  
Dim x As Integer  
Dim y as Single  
x = Val(InputBox("введи x", "ввод данных"))  
If x > 3 Then  
y = Log(x - 3) + Sin(x)  
MsgBox "x=" & x & " y=" & y  
End If  
End Sub
```

Полная форма однострочный формат:

***If* <условие> *Then* <инструкции 1> *Else* <инструкции 2>**

Здесь

*If..Then..Else* – ключевые слова, дословно означающие «Если..То..Иначе».

Данная инструкция обеспечивает выполнение группы инструкций 1, если условие выполняется, и выполнение группы инструкций 2 в противном случае.

**Задача.** Определить попадает ли точка с координатой  $(x; y)$  в круг с центром в начале координат и радиусом  $r$ .

```
Public Sub tochka()  
Dim x, y, r As Integer  
Dim s As Single  
x = Val(InputBox("введи x", "координаты точки"))  
y = Val(InputBox("введи y", "координаты точки"))
```

```

r = Val(InputBox("введи r", "радиус окружности"))
'вычисление расстояния от центра начала координат до точки
s = Sqr(x ^ 2 + y ^ 2)
If s <= r Then MsgBox "попадет" Else MsgBox "не попадет"
End Sub

```

Полная форма блочный формат:

```

If <условие> Then
<инструкции 1>
Else
<инструкции 2>
End if

```

**Задача.** Определить попадает ли точка с координатой  $(x; y)$  в круг с центром в начале координат и радиусом  $r$ .

```

Public Sub tochka()
Dim x, y, r As Integer
Dim s As Single
x = Val(InputBox("введи x", "координаты точки"))
y = Val(InputBox("введи y", "координаты точки"))
r = Val(InputBox("введи r", "радиус окружности"))
'вычисление расстояния от центра начала координат до точки
s = Sqr(x ^ 2 + y ^ 2)
If s <= r Then
MsgBox "попадет"
Else
MsgBox "не попадет"
End If
End Sub

```

#### 2.4.2 Инструкция выбора варианта

Позволяет выбрать один из  $n$  возможных вариантов решения задачи. Выбор того или иного варианта зависит от совпадения ключа с одним из предлагаемых вариантов. Данная инструкция и имеет следующий формат:

```

Select Case ключ
Case вариант 1: инструкции 1
Case вариант 2: инструкции 2
...
Case вариант n: инструкции n
Case else: инструкции case else
End Select
Здесь,

```

Ключ – переменная или выражение, значение которого нужно проверять на совпадение с одним из предлагаемых вариантов;

Вариант  $n$  – константа, диапазон констант, записываемый с помощью ключевого слова *to* (5 to 25), простые условия, записываемые с использованием операций отношения и ключевого слова *is* ( $is > 3$ ), комбинация всего

перечисленного, записанная через «запятую».

**Задача.** Определить знак введенного с клавиатуры числа  $x$ .

```
Public Sub opora()  
Dim x As Integer  
x = Val(InputBox("введи x", "ввод данных"))  
Select Case x  
Case Is < 0: MsgBox "отрицательное"  
Case Is > 0: MsgBox "положительное"  
Case Is = 0: MsgBox "ôââîî 0"  
End Select  
End Sub
```

### 2.4.3 Создание разветвляющейся функции пользователя рабочего листа

При работе в MS Excel для набора сложных формул применяется «Мастер функций», который в распоряжение пользователя предоставляет огромное количество заранее определенных встроенных в программу функций. Пользователю остается только выбрать необходимую функцию из перечня и задать для нее аргументы, прочитав их значения из соответствующих ячеек рабочего листа. VBA предоставляет пользователю инструменты для создания собственной функции, вызываемой через «Мастер функций». Такая функция должна находиться в отдельном стандартном модуле, создаваемом в среде VBA командой *Insert-Module*. В окне «Мастера функций» она будет находиться в категории *Определенные пользователем*.

**Задача.** Создать функцию пользователя, позволяющую определять, попадает ли точка с координатой  $(x; y)$  в круг с центром в начале координат и радиусом  $r$ . Затем решить задачу с использованием созданной функции на листе 2 текущей рабочей книги, задавая произвольным образом координаты рассматриваемых точек и радиусы предполагаемых окружностей.


#### Порядок решения задачи:

1 Запустить редактор VBE, нажав *Alt+F11* или кнопку *Visual Basic* на вкладке Разработчик.

2 Создать новый модуль, выполнив команду *Insert – Module*.

3 Сформировать программный код функции

```
Public Function tochka(x As Integer, y As Integer, r As Integer) As Variant  
Dim s As Single  
s = Sqr(x ^ 2 + y ^ 2)  
If s <= r Then tochka = "попадет" Else tochka = "не попадет"  
End Function
```

4 Вернуться в окно программы MS Excel, нажав кнопку  на панели инструментов.

5 Ввести исходные данные для решения задачи (значения  $x, y, r$ ).

	A	B	C
1	$x$	$y$	$r$
2	2	3	4
3	-4	6	5
4	4	7	15
5	8	-9	5

6 Вычислить результат для первой группы значений. Для этого установить курсор в ячейку D2, вызвать “Мастер функций”, выбрать категорию “Определенные пользователем”, выбрать функцию с именем *tochka*, в появившемся окне указать адреса ячеек с исходными данными.

7 Вычислить остальные значения, используя маркер автозаполнения. Результаты в режиме отображения формул и значений показаны ниже.

	A	B	C	D
1	$x$	$y$	$r$	результат
2	2	3	4	попадет
3	-4	6	5	не попадет
4	4	7	15	попадет
5	8	-9	5	не попадет

Режим значений

	A	B	C	D
1	$x$	$y$	$r$	результат
2	2	3	4	=tochka(A2;B2;C2)
3	-4	6	5	=tochka(A3;B3;C3)
4	4	7	15	=tochka(A4;B4;C4)
5	8	-9	5	=tochka(A5;B5;C5)

Режим формул

#### 2.4.4 Макросы

Программу на языке VBA можно создавать двумя способами:

1 Автоматически, при помощи макрорекодера. **Макрорекодер** – транслятор действий пользователя, которые записываются в виде соответствующих команд в программу на VBA, называемую **макросом**.

2 Формируя программу вручную, используя возможности среды программирования VBE.



В данной контрольной работе макросы будут использоваться для осуществления форматирования текстовой части работы в MS Word. Создавать макросы будем при помощи макрорекодера. Поэтому рассмотрим процесс создания макроса именно этим способом.

В первую очередь на ленту необходимо вывести вкладку Разработчик. Для этого необходимо выполнить команду *Файл – Параметры – Настройка ленты*, и активировать флажок рядом с именем вкладки «Разработчик».

На вкладке Разработчик выполнить команду *Безопасность макросов* и установить переключатель в положение «Включить все макросы».

Записать в макрос требуемые команды. Для этого нужно совершить следующие действия:

- 1 Выполнить команду *Запись макроса* на вкладке Разработчик.
- 2 В строке «Имя макроса» ввести имя создаваемого макроса, а из списка «Макрос доступен для» выбрать текущий документ.
- 3 Далее нажать кнопку *Панели*.
- 4 Выделить создаваемый макрос и нажать кнопку *Добавить*.
- 5 Назначить иконку для кнопки макроса и дать ему краткое имя. Для этого нажать кнопку *Изменить* и выполнить необходимые манипуляции.
- 6 Записать в макрос команды, выбирая их из соответствующих диалоговых окон или нажимая командные кнопки на панелях.
- 7 После записи всех команд нажать кнопку *Остановить запись*.



*Примечание* – В макрос записываются все действия, которые выполняет пользователь в процессе его записи. Это означает, что во время записи макроса нельзя выполнять какие-либо другие команды, не относящиеся к командам, которые нужно записать в него. Чтобы временно приостановить запись макроса используйте команду Пауза.

### 3 РАСЧЕТ СТАТИЧЕСКИ ОПРЕДЕЛИМОЙ ДВУТАВРОВОЙ БАЛКИ

Балки предназначены для восприятия поперечных нагрузок. По способу приложения нагрузки делятся на две группы:

- сосредоточенные, которые действуют на точку;
- распределенные, которые действуют на значительную площадь или длину.

Балки имеют опорные устройства для сопряжения их с другими элементами и передачи на них усилий. Применяются следующие виды опор:

- 1 Шарнирно-подвижная . Эта опора допускает поворот вокруг оси и линейное перемещение параллельно опорной плоскости.
- 2 Шарнирно-неподвижная . Эта опора допускает поворот вокруг оси и не допускает линейное перемещение параллельно опорной плоскости.
- 3 Жесткая заделка. Эта опора не допускает поворотов и перемещений.

Двухопорной называют балку, расположенную на двух шарнирных опорах. При этом одна из опор должна быть шарнирно-неподвижной (точка  $B$ ), а другая шарнирно-подвижной (точка  $A$ ).



Рисунок 9 – Двухопорная балка

Часть балки расположенная между смежными опорами называется пролетом. Выступающие части балки называются консолями.

Расчет балки на прочность, расположенной на двух шарнирных опорах сводится к прохождению следующих этапов:

1 Определение реакции опор  $R_a$  и  $R_b$  из следующих уравнений статики:

$$\sum M_a = 0$$

$$\sum M_b = 0$$

2 Выбор характерных сечений и определение поперечных сил  $Q_x$  в них.

Построение эпюры поперечных сил.

3 Определение изгибающих моментов в характерных сечениях  $M_x$ . Построение эпюры изгибающих моментов.

4 Подбор сечения двутавровой балки, исходя из условия:

$$W_x \geq \frac{M_{\max}}{\sigma}$$

5 Проверка сечения балки по касательным напряжениям, исходя из условия

$$\tau_{\max} < \tau.$$

$$\tau_{\max} = \frac{Q_{\max} \cdot S_{\max}}{b_0 \cdot I}.$$

Здесь,

$Q_{\max}$  – максимальная поперечная сила;

$S_{\max}$  – статический момент;

$b_0 = d$  – толщина стенки двутавра;

$I$  – момент инерции относительно оси  $Ox$ .

6 Вычисление максимального нормального напряжения в крайнем волокне опасного сечения  $\sigma_{\max} = \frac{M_{\max}}{W_x}$  и проверка условия:  $\sigma_{\max} \leq \sigma$ .

#### 4 ПРИМЕР ВЫПОЛНЕНИЯ ЗАДАНИЙ КОНТРОЛЬНОЙ РАБОТЫ

**Задача.** Произвести расчет на прочность статически определимой двутавровой балки (рисунок 10) при следующих данных:  $F=50\text{кН}$ ,  $q=40\text{ кН/м}$ ,  $a=0,7\text{ м}$ ,  $l=3\text{м}$ , допустимые нормальные и касательные напряжения:  $\sigma=160\text{ МПа}$  и  $\tau=100\text{ МПа}$ .

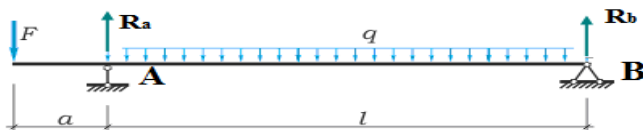


Рисунок 10 – Статически определимая двутавровая балка

##### Порядок решения задачи:

1 Определить опорные реакции по формулам:

$$R_a = \frac{F \cdot (a + L) + \frac{q \cdot L^2}{2}}{L} \quad R_b = \frac{\frac{q \cdot L^2}{2} - F \cdot a}{L}$$

2 Построить эпюры поперечных сил  $Q$  и изгибающего момента  $M$ . Для расчета  $Q$  и  $M$  использовать формулы:

$$Q(x) = \begin{cases} -R_b + q \cdot L - R_a, & \text{если } L \leq x \leq L + a \\ -R_b + q \cdot x, & \text{если } 0 \leq x \leq L \end{cases}$$

$$M(x) = \begin{cases} R_b \cdot x - q \cdot L \cdot \left(x - \frac{L}{2}\right) + R_a \cdot (x - L), & \text{если } L \leq x \leq L + a \\ R_b \cdot x - q \cdot \frac{x^2}{2}, & \text{если } 0 \leq x \leq L \end{cases}$$

3 Вычислить максимальный изгибающий момент  $M_{\max}$ .

4 Подобрать сечение двутавровой балки  $W_x$ , пользуясь сортаментом

(приложение А).  $W_x \geq \frac{M_{\max}}{\sigma}$

5 Вычислить  $\sigma_{\max} = \frac{M_{\max} \cdot 10^3}{W_x}$ , и убедиться, что  $\sigma_{\max} \leq \sigma$ , выполнив проверку балки на прочность по нормальным напряжениям.

6 Вычислить  $\tau_{\max} = \frac{Q_{\max} \cdot S_{\max}}{b_0 \cdot I} \cdot 10^2$ ,  $b_0 = d$ , и убедиться, что  $\tau_{\max} \leq \tau$ , выполнив проверку балки на прочность по касательным напряжениям.

**Задание 1.** Используя исходные данные, сформировать документ *MS Excel*, в котором выполняются необходимые вычисления. Для вычисления

$R_a$ ,  $R_b$  составить, используя VBA, функции пользователя и применить их для выполнения соответствующих вычислений на рабочем листе. Для вычисления значений  $Q_x$  и  $M_x$  использовать функцию ЕСЛИ. Для проверки выбранной балки на прочность составить, используя VBA, функции пользователя и применить их для выполнения соответствующих вычислений. Исходные данные и результаты расчетов должны быть оформлены в виде таблиц с границами. Диаграмма должна размещаться на отдельном листе.

### Порядок выполнения задания 1:

- 1 Загрузить MS Excel.
- 2 На первом листе рабочей книги сформировать таблицу с исходными данными:

	A	B	C	D	E
1					
2		Исходные данные:			
3		Сосредоточенная нагрузка	F, кН	50	
4		Распределенная нагрузка	q, кН	40	
5		Длина консоли	a, м	0,7	
6		Длина пролета	l, м	3	
7		Допустимое нормальное напряжение	$\sigma$ , Мпа	160	
8		Допустимое касательное напряжение	$\tau$ , Мпа	100	

- 3 Вычислить опорные реакции по формулам:

$$R_a = \frac{F \cdot (a + L) + \frac{q \cdot L^2}{2}}{L} \quad R_b = \frac{\frac{q \cdot L^2}{2} - F \cdot a}{L}$$

Для вычисления опорных реакций сначала составить на VBA две линейные функции, а затем использовать их для выполнения соответствующих вычислений, осуществляя их вызов “Мастером функций”. Для формирования программного кода функции  $R_a$ :

- загрузите среду программирования, нажав  $Alt+F11$ ;
- создайте новый модуль, выполнив команду *Insert – Module*;
- наберите программный код функции согласно образцу:

```
Public Function Ra(F As Integer, q As Integer, a As Single, l As Single) As Single
Ra = (F * (a + l) + q * l ^ 2 / 2) / l
End Function
```

Результаты вычислений оформить в таблицу:

	A	B	C	D	E
9					
10		Определение опорных реакций			
11		$R_a$	121,6667		
12		$R_b$	48,33333		

4 Вычислить поперечные силы  $Q_x$  и изгибающие моменты  $M_x$  в характерных сечениях. Для вычисления  $Q_x$  и  $M_x$  использовать функцию ЕСЛИ.

Формат функции ЕСЛИ для вычисления  $Q_x$  в точке  $x=3.7$  показан ниже:

=ЕСЛИ(И(В16<=\$D\$6+\$D\$5;В16>=\$D\$6;ЯЧЕЙКА("содержимое";В16)<>ЯЧЕЙКА("содержимое";В15));-\$C\$12+\$D\$4\*\$D\$6-\$C\$11;ЕСЛИ(И(В16<=\$D\$6;В16>=0);-\$C\$12+\$D\$4\*В16))

*Примечания*

- 1 Функция ЕСЛИ записывается в одну строку;
- 2 Для ссылок на ячейки, содержащие постоянные значения используется абсолютный адрес (\$);
- 3 Функция ЯЧЕЙКА сравнивает значения в соседних ячейках;
- 4 Значения переменной  $x$  задаются с шагом 0,1 в два этапа: сначала вводится диапазон  $l \leq x \leq l + a$ , а затем  $0 \leq x \leq l$ .
- 5 Значения переменной  $x$  вводить в порядке убывания.

Результаты вычислений оформить в таблицу, фрагмент которой показан ниже:

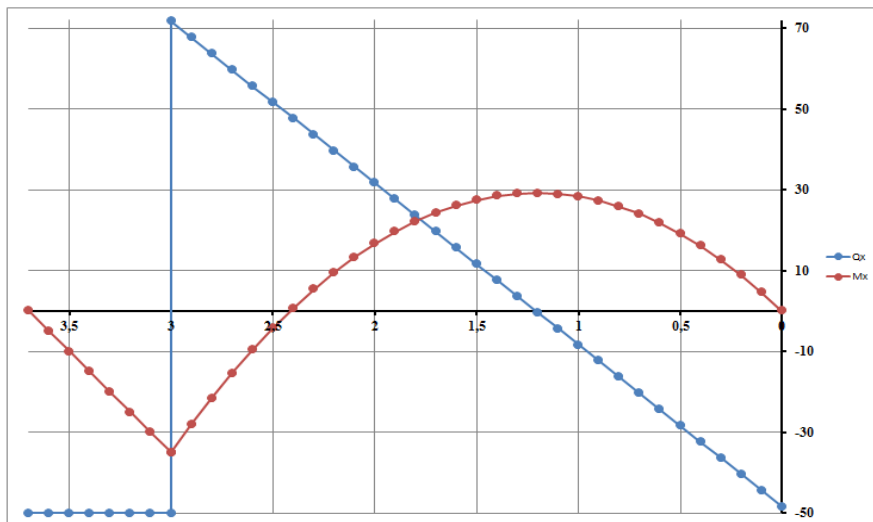
	A	B	C	D	E
13					
14					
15		x	Q <sub>x</sub>	M <sub>x</sub>	
16			3,7	-49,999996	-6,485E-06
17			3,6	-49,999996	-5,0000061
18			3,5	-49,999996	-10,000006
19			3,4	-49,999996	-15,000005
20			3,3	-49,999996	-20,000005
21			3,2	-49,999996	-25,000005
22			3,1	-49,999996	-30,000004
23			3	-49,999996	-35,000004
24			3	71,666668	-35,000004
25			2,9	67,666668	-28,033337

5 Построить эпюры поперечных сил  $Q$  и изгибающего момента  $M$  в одной координатной плоскости. Для корректного построения выделить три столбца со значениями  $x, Q_x, M_x$  и выполнить команду Вставка – Точечная (с прямыми отрезками и маркерами). Отредактировать построенную диаграмму:

- разместить диаграмму на отдельном листе, выполнив команду Конструктор – Переместить диаграмму – на отдельном листе;
- настроить обратный порядок нумерации по оси  $Ox$ , а также минимальное и максимальное значение на этой же оси, толщину (3 пт) и цвет (черный) линии оси, используя команды диалога Макет – Оси – Основная горизонтальная ось – Дополнительные параметры;

- аналогично настроить параметры вертикальной оси;
- задать основные линии горизонтальной и вертикальной сетки, используя соответствующие команды, находящиеся по адресу *Макет – Сетка*;
- изменить шрифт нумерации осей: тип – *Times New Roman*, размер – 12 пт, начертание – полужирный.

Построенные эпюры показаны на образце ниже



6 Вычислить максимальный изгибающий момент  $M_{max}$  и максимальную поперечную силу  $Q_{max}$ , используя стандартную функцию *MS Excel*.

	A	B	C	D	E
55					
56		$M_{max}$	29,199998		
57		$Q_{max}$	71,666668		

7 Подобрать сечение двутавровой балки  $W_x$ , пользуясь сортаментом “Двутавры стальные горячекатаные (ГОСТ 8239-89)” (см. Приложение А) так, чтобы неравенство  $W_x \geq \frac{M_{max}}{\sigma}$  выполнялось. Для этого сначала рассчита-

ть  $W_{max} = \frac{M_{max} \cdot 10^3}{\sigma}$ , а затем по сортаменту подобрать необходимый двутавр, и выписать его характеристики, необходимые для дальнейших вычислений.

	A	B	C	D	E
58					
59		Подбор сечения			
60		$W_{max}$	182,49999	см <sup>3</sup>	
61					
62		Взяли двугавр №20 со следующими параметрами:			
63		$W_x$	184	см <sup>3</sup>	
64		$I_x$	1840	см <sup>4</sup>	
65		$S_{max}$	104	см <sup>3</sup>	
66		$d=b_0$	5,6	мм	

8 Вычислить  $\sigma_{max} = \frac{M_{max} \cdot 10^3}{W_x}$  и  $\tau_{max} = \frac{Q_{max} \cdot S_{max}}{b_0 \cdot I} \cdot 10^2$

9 Убедиться, что  $\sigma_{max} \leq \sigma$ , выполнив проверку балки на прочность по нормальным напряжениям.

10 Убедиться, что  $\tau_{max} \leq \tau$ , выполнив проверку балки на прочность по касательным напряжениям.

11 Для выполнения п.8 и п. 9 составить программный код двух разветвляющихся функций, которые проверяют правильность выполненного расчета, исходя из условий прочности балки по нормальным и касательным напряжениям. Функции должны выдавать на экран одно из двух текстовых сообщений: “подходит”, если условие прочности выполняется и “не подходит” – в противном случае, и вызываться из диалога “*Мастер функций*”. Как и раньше каждая функция должна находиться в своем модуле.

Пример программного кода функции для проверки прочности по нормальным напряжениям с использованием условной инструкции *if..then..else*:

```
Public Function sigma_rez(sigmamax As Single, sigma As Single) As Variant
If sigmamax <= sigma Then
sigma_rez = "подходит"
Else
sigma_rez = "не подходит"
End If
End Function
```

Пример программного кода функции для проверки прочности по нормальным напряжениям с использованием инструкции выбора варианта *select..case*:

```
Public Function sigma_rez(sigmamax As Single, sigma As Single) As Variant
Select Case sigmamax
Case Is <= sigma: sigma_rez = "подходит"
Case Else: sigma_rez = "не подходит"
End Select
End Function
```

	A	B	C	D	E
67					
68		Проверка на прочность			
69		по нормальным напряжениям, $\sigma_{max}$	158,6956	МПа	подходит
70		по касательным напряжениям, $\tau_{max}$	72,33437	МПа	подходит

## 12 Сохранить файл с именем **Контрольная 2\_ФИО**

**Задание 2.** Записать, а затем выполнить макрос, осуществляющий форматирование текста введения, заключения и списка литературы. Макрос должен выполняться при нажатии на кнопку, созданную на панели инструментов. Параметры форматирования, записываемые в макрос: тип шрифта - *Times New Roman*, размер - 9, начертание - курсивное, выравнивание текста - по ширине, красная строка - 2 см, междустрочный интервал - полуторный, расстановка переносов - есть.

### Порядок выполнения задания 2:

- 1 Загрузить MS Word
- 2 Сформировать отчет к контрольной работе, структура которого описана в задании 3.
- 3 Вывести на ленту вкладку Разработчик, выполнив команду *Файл – Параметры – Настройка ленты* (активировать флажок *Разработчик*).
- 4 Выполнить команду *Безопасность макросов* на вкладке Разработчик и выбрать опцию *Включить все макросы*.
- 5 Выполнить команду *Запись макроса*.
- 6 В первом окне указать его имя «Контрольная», область доступности «в текущем документе», а также назначить его выполнение кнопке на панели: нажать кнопку *Панели*, далее выбрать из списка создаваемый макрос и нажать на кнопку *Добавить*. Настроить текст кнопки и иконку, нажав кнопку *Изменить* в текущем окне, выполнив далее требуемые операции. Далее нажать ОК. При этом макрос будет выполняться при нажатии на кнопку, созданную на панели быстрого запуска.
- 7 Записать необходимые команды в макрос, просто выбирая их на панелях, или найдя аналоги в соответствующих диалоговых окнах. При записи макроса нельзя переключаться из окна в окно. Все действия, выполняемые пользователем в процессе записи макроса, в него записываются. Поэтому можно использовать кнопку *Пауза*, чтобы приостановить запись команд в макрос.
- 8 По окончании записи команд нажать на кнопку *Остановить запись*.
- 9 Применить макрос к тексту введения, заключения, а также к списку использованных источников. Для этого выделить нужный фрагмент и нажать кнопку, запускающую макрос.



## Структура итогового отчета.

Итоговый отчет должен иметь следующую структуру:

- 1 Титульный лист (приложение Б).
- 2 Задание на контрольную работу (приложение В), выданное и подписанное преподавателем.
- 3 Содержание (составляется автоматически).
- 4 Введение. В данном разделе описываются цели и задачи контрольной работы. Объем – 0,5 с.
- 5 Раздел **Расчет на прочность двутавровой балки** состоит из подразделов:
  - *Результаты в режиме отображения значений*. Содержит результаты расчета в режиме отображения значений.
  - *Результаты в режиме отображения формул*. Содержит результаты расчета в режиме отображения формул.
  - *Функции пользователя*. Содержит программный код четырех функций пользователя.
  - *Диаграмма*. Содержит построенную диаграмму.
- 6 Раздел **Макрос** содержит программный код макроса.
- 7 Заключение. В данном разделе формируются выводы по результатам выполненной работы.
- 8 Список использованной литературы.

**Задание 3.** Отформатировать отчет согласно требованиям.

**Порядок выполнения задания 3.** Не закрывая файл, в котором записали макрос, сформируйте и отформатируйте отчет согласно требованиям, а затем сохраните файл с именем **Контрольная 2\_ФИО**

## 5 ТРЕБОВАНИЯ К ФОРМАТИРОВАНИЮ ИТОГОВОГО ОТЧЕТА

1 Каждый основной раздел должен начинаться с новой страницы и иметь заголовок, выполненный стилем *Заголовок 1*, со следующими параметрами: размер шрифта – 14; тип шрифта – *Times New Roman*; начертание – полужирное; выравнивание текста – по центру. К основным разделам, кроме описанных выше, относят *Введение*, *Заключение*, *Список литературы*.

2 Названия подразделов основных разделов форматируются стилем *Заголовок 2*, со следующими параметрами: размер шрифта – 13; тип шрифта – *Times New Roman*; начертание шрифта – полужирный курсив; выравнивание текста – по центру.

3 В верхний колонтитул документа необходимо по центру ввести Ф.И.О, группу и учебный шифр.

4 Страницы документа должны быть пронумерованы. Номер страницы вводится в нижний колонтитул по правому краю и форматируется шрифтом размером – 13. Титульный лист не нумеруется, хотя и включается в общую нумерацию страниц документа.

5 В документе должны быть установлены следующие поля: левое – 30 мм; правое – 10 мм; верхнее – 15 мм; нижнее – 15 мм.

## СПИСОК ЛИТЕРАТУРЫ

- 1 *Алиев, В. Excel 2010. Проще простого* / В. Алиев. – СПб. : Питер, 2007. – 144 с.
- 2 *Кашаев, С. Офисные решения с использованием Microsoft Excel 2007 и VBA* / С. Кашаев. – СПб. : Питер, 2009. – 352 с.
- 3 *Кузьменко, В. Г. VBA. Эффективное использование* / В. Г. Кузьменко. – М. : Бином, 2012. – 624 с.
- 4 Расчет балки на прочность и жесткость. [Электронный ресурс]. – Режим доступа: <http://www.isopromat.ru/sopromat/primery-reshenia-zadach/rascet-balki-na-procnost-i-zestkost>. – Дата доступа : : 15.11.2015.

**ПРИЛОЖЕНИЕ А**

(обязательное)

Сортамент двутавров ГОСТ 8239 (сокращенная версия)

№ балки	Размеры, мм				Площадь сечения, см <sup>2</sup>	Справочные величины для оси Oх		
	<i>h</i>	<i>b</i>	<i>d</i>	<i>t</i>		<i>I<sub>x</sub></i> , см <sup>4</sup>	<i>W<sub>x</sub></i> , см <sup>3</sup>	<i>S<sub>x</sub></i> , см <sup>3</sup>
10	100	55	4,5	7,2	12,0	198	39,7	23,0
12	120	64	4,8	7,3	14,7	350	58,4	33,7
14	140	73	4,9	7,5	17,4	572	81,7	46,8
16	160	81	5,0	7,8	20,2	573	109,0	62,3
18	180	90	5,1	8,1	23,4	1290	143,0	81,4
18а	180	100	5,1	8,3	25,4	1430	159,0	89,8
20	200	100	5,2	8,4	26,8	1840	184,0	104,0
20а	200	110	5,2	8,6	28,9	2030	203,0	114,0
22	220	110	5,4	8,7	30,6	2550	232,0	131,0
22а	220	120	5,4	8,9	32,8	2790	254,0	143,0
24	240	115	5,6	9,5	34,8	3460	289,0	163,0
24а	240	125	5,6	9,8	37,5	3800	317,0	178,0
27	270	125	6,0	9,8	40,2	5010	371,0	210,0
27а	270	135	6,0	10,2	43,2	5500	407,0	229,0
30	300	135	6,5	10,2	46,5	7080	472,0	268,0
30а	300	145	6,5	10,7	49,9	7780	518,0	292,0
33	330	140	7,0	11,2	53,8	9840	597,0	339,0
36	360	145	7,5	12,3	61,9	13380	743,0	423,0
40	400	155	8,3	13,0	72,6	19062	953,0	545,0
45	450	160	9,0	14,2	84,7	27696	1231,0	708,0
50	500	170	10,0	15,2	100,0	39727	1589,0	919,0
55	550	180	11,0	16,5	118,0	55962	2035,0	1181,0
60	600	190	12,0	17,8	138,0	76806	2560,0	1491,0

*ПРИЛОЖЕНИЕ Б*  
*(обязательное)*  
**Титульный лист (образец)**

МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ  
РЕСПУБЛИКИ БЕЛАРУСЬ  
УО «Белорусский государственный университет транспорта»  
Кафедра «Информационные технологии»

КОНТРОЛЬНАЯ РАБОТА №2  
по дисциплине «Информатика»  
на тему: «Расчет статически определимой двутавровой балки на прочность  
средствами *VBA* и *MS Excel*»

Выполнил:  
студент группы ЗП-11  
Иванов И.И  
учебный шифр 456

Проверил:

Гомель 2016

## ПРИЛОЖЕНИЕ В

(обязательное)

### Бланк задания на контрольную работу №2

МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ  
РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Белорусский государственный университет транспорта»  
Кафедра «Информационные технологии»

#### Задание

#### на контрольную работу №2

тема: “ Расчет статически определимой двутавровой балки на прочность средствами *VBA* и *MS Excel*”

по дисциплине “Информатика” для студентов 2 курса заочного факультета специальности П.

СТУДЕНТУ \_\_\_\_\_ ГРУППЫ \_\_\_\_\_

**Задача.** Произвести расчет на прочность статически определимой двутавровой балки. Исходные данные взять из **таблицы Б.1** по *последней цифре* шифра.

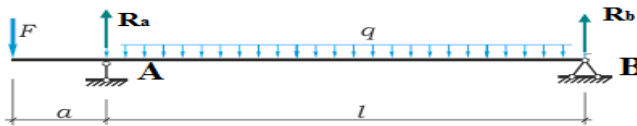


Таблица Б.1– Выбор варианта для решения задачи

Номер варианта	Сосредоточенная нагрузка $F$ , кН	Распределенная нагрузка $q$ , кН	Длина консоли $a$ , м	Длина пролета $L$ , м	Допустимое нормальное напряжение $\sigma$ , МПа	Допустимое касательное напряжение $\tau$ , МПа
0	40	30	0,8	4	160	100
1	40	30	1	4	190	100
2	35	50	0,8	4	160	100
3	35	45	1	3,8	140	100
4	35	50	1	4,2	200	100
5	40	35	1	3,8	160	100
6	50	45	1,2	5	160	100
7	35	25	0,6	3,6	160	100
8	45	35	0,8	3,6	170	100
9	50	40	1,2	4,4	150	100

#### Порядок решения задачи:

- 1 Определить опорные реакции по формулам:

$$R_a = \frac{F \cdot (a + L) + \frac{q \cdot L^2}{2}}{L}$$

$$R_b = \frac{\frac{q \cdot L^2}{2} - F \cdot a}{L}$$

2 Построить эпюры поперечных сил  $Q$  и изгибающего момента  $M$ . Для расчета  $Q$  и  $M$  использовать формулы:

$$Q(x) = \begin{cases} -R_b + q \cdot L - R_a, & \text{если } L \leq x \leq L + a \\ -R_b + q \cdot x, & \text{если } 0 \leq x \leq L \end{cases}$$

$$M(x) = \begin{cases} R_b \cdot x - q \cdot L \cdot \left(x - \frac{L}{2}\right) + R_a \cdot (x - L), & \text{если } L \leq x \leq L + a \\ R_b \cdot x - q \cdot \frac{x^2}{2}, & \text{если } 0 \leq x \leq L \end{cases}$$

3 Вычислить максимальный изгибающий момент  $M_{\max}$ .

4 Подобрать сечение двутавровой балки  $W_x$ , пользуясь сортаментом

(см. Приложение А).  $W_x \geq \frac{M_{\max}}{\sigma}$

5 Вычислить  $\sigma_{\max} = \frac{M_{\max} \cdot 10^3}{W_x}$ , и убедиться, что  $\sigma_{\max} \leq \sigma$ , выполнив проверку балки на прочность по нормальным напряжениям.

6 Вычислить  $\tau_{\max} = \frac{Q_{\max} \cdot S_{\max}}{b_0 \cdot I} \cdot 10^2$ , и убедиться, что  $\tau_{\max} \leq \tau$ , выполнив проверку балки на прочность по касательным напряжениям.

**Задание 1.** Используя исходные данные, сформировать документ *MS Excel*, в котором выполняются необходимые вычисления. Для вычисления  $R_a$ ,  $R_b$  составить, используя *VBA*, функции пользователя и применить их для выполнения соответствующих вычислений на рабочем листе. Для вычисления значений  $Q_x$  и  $M_x$  использовать функцию ЕСЛИ. Для проверки выбранной балки на прочность по нормальным и касательным напряжениям составить, используя *VBA*, функции пользователя и применить их для выполнения соответствующих вычислений. Для программирования требуемых функций использовать **оператор IF**, если последняя цифра четная или **оператор Select Case**, если последняя цифра не четная. Исходные данные и результаты расчетов должны быть оформлены в виде таблиц с границами. Диаграмма должна размещаться на отдельном листе.

#### Примечания

1 Расчетные значения  $x$  задавать в обратном порядке от  $L+a$  до 0 с шагом 0,1 в два этапа:  $[L+a; L]$  и  $[L; 0]$ .

2 Построить эпюры поперечных сил и изгибающего момента в одной координатной плоскости. Для корректного построения выбрать тип диаграммы «точечная».

**Задание 2.** MS Word. Составить отчет, структура которого описана ниже. Записать, а затем выполнить макрос, осуществляющий форматирование текста введения, заключения и списка литературы. Макрос должен выполняться при нажатии на кнопку, созданную на панели быстрого запуска. Вариант задания взять из **таблицы Б.2** по *предпоследней цифре* шифра.

**Таблица Б.2– Выбор варианта для создания макроса**

Номер варианта	Задание
0	Шрифт - <i>Times New Roman</i> , размер -10, начертание – полужирное, выравнивание – по ширине, красная строка – 0,5 см, полуторный междустрочный интервал, расстановка переносов.
1	Шрифт - <i>Arial</i> , размер -9, начертание – курсив, выравнивание – по ширине, красная строка – 0,75 см, одинарный междустрочный интервал, расстановка переносов
2	Шрифт - <i>Tahoma</i> , размер -9,5, начертание – курсив, выравнивание – по ширине, красная строка – 1 см, двойной междустрочный интервал.
3	Шрифт - <i>Times New Roman</i> , размер -11, начертание – полужирный курсив, выравнивание – по ширине, красная строка – 0,65 см, полуторный междустрочный интервал, расстановка переносов.
4	Шрифт - <i>Courier</i> , размер -10,5, начертание – курсив, выравнивание – по ширине, красная строка – 1,25 см, одинарный междустрочный интервал, расстановка переносов
5	Шрифт - <i>Tahoma</i> , размер -8,5, начертание – курсив, выравнивание – по ширине, красная строка – 0,85 см, двойной междустрочный интервал, расстановка переносов.
6	Шрифт - <i>Arial</i> , размер -10, начертание – полужирный, выравнивание – по ширине, красная строка – 1,15 см, полуторный междустрочный интервал, расстановка переносов
7	Шрифт - <i>Times New Roman</i> , размер - 12, начертание – полужирный, выравнивание – по ширине, красная строка – 1,2 см, двойной междустрочный интервал.
8	Шрифт - <i>Bookman Old Style</i> , размер -9, начертание – курсив, выравнивание – по ширине, красная строка – 2 см, одинарный междустрочный интервал, расстановка переносов
9	Шрифт - <i>Tahoma</i> , размер -8, начертание – полужирный курсив, выравнивание – по ширине, красная строка – 1,3 см, двойной междустрочный интервал.

Итоговый отчет должен иметь следующую структуру:

- 1 Титульный лист (приложение Б).
- 2 Задание на контрольную работу (приложение В), выданное и подписанное преподавателем.
- 3 Содержание (составляется автоматически).

- 4 Введение. В данном разделе описываются цели и задачи контрольной работы. Объем – 0,5 с.
- 5 Раздел **Расчет на прочность двутавровой балки** состоит из подразделов:
  - *Результаты в режиме отображения значений*. Содержит результаты расчета в режиме отображения значений.
  - *Результаты в режиме отображения формул*. Содержит результаты расчета в режиме отображения формул.
  - *Функции пользователя*. Содержит программный код четырех функций пользователя.
  - *Диаграмма*. Содержит построенную диаграмму.
- 6 Раздел **Макрос** содержит программный код макроса.
- 7 Заключение. В данном разделе формируются выводы по результатам выполненной работы.
- 8 Список использованной литературы.

**Задание 3.** Отформатировать отчет согласно требованиям.

### **Требования к форматированию итогового отчета**

- 1 Каждый основной раздел должен начинаться с новой страницы и иметь заголовки, выполненный стилем *Заголовок 1*, со следующими параметрами: размер шрифта – 14; тип шрифта – *Times New Roman*; начертание – полужирное; выравнивание текста – по центру. К основным разделам, кроме описанных выше, относят *Введение*, *Заключение*, *Список литературы*.
- 2 В верхний колонтитул документа необходимо по центру ввести Ф.И.О, группу и учебный шифр.
- 3 Страницы документа должны быть пронумерованы. Номер страницы вводится в нижний колонтитул по правому краю и форматируется шрифтом размером – 13. Титульный лист не нумеруется, хотя и включается в общую нумерацию страниц документа.
- 4 В документе должны быть установлены следующие поля: левое – 30 мм; правое – 10 мм; верхнее – 15 мм; нижнее – 15 мм.

### **Рекомендуемая литература**

- 1 Пшеничнов Ю. А. Информатика: практикум / Ю. А. Пшеничнов; БелГУТ – Гомель : 2008. – 343
- 2 Борисенко М. В. Компьютерные информационные технологии: учеб. – метод. пособие / М. В. Борисенко ; БелГУТ – Гомель : 2008. – 29
- 3 БелГУТ, кафедра “Информационные технологии” [Электронный ресурс] / Белорус. гос. ун-т трансп. – Гомель, 2008–2016. – Режим доступа : <http://www.zf.belsut.gomel.by/it>. – Дата доступа : 10.03.2016.



Учебное издание

*ЛИТВИНОВИЧ Татьяна Николаевна*

ПРИМЕНЕНИЕ ВВА ДЛЯ РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ

Учебно-методическое пособие  
по выполнению контрольной работы № 2

Редактор *А. А. Павлюченкова*  
Технический редактор *В. Н. Кучерова*  
Корректор *Т. А. Пугач*

Подписано в печать 26.04.2016 г. Формат 60x84 <sup>1</sup>/<sub>16</sub>.  
Бумага офсетная. Гарнитура Times. Печать на ризографе.  
Усл. печ. л. 2,32. Уч.-изд. л. 2,18. Тираж 200 экз.  
Зак. 1556. Изд. № 125

Издатель и полиграфическое исполнение:  
Белорусский государственный университет транспорта.  
Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий  
№ 1/361 от 13.06.2014г.  
№ 2/104 от 01.04.2014г.  
Ул. Кирова, 34, 246653, Гомель